CS 134: Networks	Problem set 4
Prof. Yaron Singer	Due: **Wednesday 3/1/2017**

Resources:

This problem set contains 5 parts.

1. Centrality (Jackson 2.2.4). The *degree centrality* of a node is its degree divided by $n-1$ (where n is the number of nodes in the network). For example, the degree centrality of node d in network A is $\frac{3}{5}$.

a. Compute the degree centrality of each of the nodes in network B.

Closeness Centrality. The most basic closeness centrality measure is the inverse of the average distance between i and all other nodes j in the network[1]. To illustrate, let us compute the closeness centrality of node b in the following network

$$(\{a, b, c\}, \{ab, ac\}).$$

- The distance between node b and node a is 1.
- The distance between node b and node c is 2.

The average of these distances is $\frac{3}{2}$, so the closeness centrality of node b in this network is $\frac{2}{3}$.

b. Compute the closeness centrality of node d in network

$$(\{a, b, c, d\}, \{ab, ac, ad, bc, cd\}).$$

Solution by Olivia Chung:

a.

$$a, b, c, d, h, l = \frac{3}{11}$$
$$e, f, g, j, k, i = \frac{2}{11}$$

b.

- The distance between node d and node a is 1.
- The distance between node d and node b is 2.
- The distance between node d and node c is 1.

The average of these distances is $\frac{4}{3}$, so the closeness centrality of node d in this network is $\frac{3}{4}$.

[1]There are various conventions for handling networks that are not connected, as well as other possible measures of distance, which lead to a family of closeness measures, but for now we won't get into this.

2. Betweenness Centrality. Let $P_i(k,j)$ denote the number of geodesics between nodes k and j that node i lies on, and let $P(k,j)$ be the total number of geodesics between nodes k and j. We can estimate how important node i is in terms of connecting nodes k and j by looking at the fraction $\frac{P_i(k,j)}{P(k,j)}$ of shortest paths between nodes k and j that involve node i. Averaging this ratio across all pairs of nodes gives us the *betweeness centrality* of node i. To illustrate, let us compute the betweenness centrality of node c in the following "circle network"

$$(\{a,b,c,d\},\{ab,bc,cd,da\}).$$

- None of the shortest paths between node a and node b involve node c, so $\frac{P_c(a,b)}{P(a,b)} = 0$.
- None of the shortest paths between node a and node d involve node c, so $\frac{P_c(a,d)}{P(a,d)} = 0$.
- One of the two shortest paths between nodes b and node d involve node c, so $\frac{P_c(b,d)}{P(b,d)} = \frac{1}{2}$.

Averaging out these three ratios, we find that the betweenness centrality of node c in this network is $\frac{1}{6}$.

Compute the betweenness centrality of each of the nodes in the network $(\{a,b,c,d,e,f\},\{ab,bc,cd,ce,df,ef\})$.

Solution by Richard Ouyang:

- It's clear by drawing the graph that no shortest path not involving a goes through a, so its betweenness centrality is 0.
- Of all the shortest paths not involving b, only the ones that involve a go through b. There are 10 pairs, 4 of which involve a, so b's betweenness centrality is $4/10 = 2/5$.
- Of the 10 shortest paths not involving c, only ab, df, ef don't involve c. de involves c half the time. All others always involve c, so c's betweenness centrality is $6.5/10 = 13/20$.
- Of the 10 shortest paths not involving d, only af, bf, cf involve d, each of which does so half the time. All others never involve d, so d's betweenness centrality is $3/20$.
- Of the 10 shortest paths not involving e, only af, bf, cf involve e, each of which does so half the time. All others never involve e, so e's betweenness centrality is $3/20$.
- Of the 10 shortest paths not involving f, only de involves f, which does so half the time. All others never involve f, so f's betweenness centrality is $1/20$.

3. Programming: Verifying the Friendship Paradox. Using the eight sample directed social networks below, courtesy of SNAP (https://snap.stanford.edu/data), for each social network, compute the value of $\frac{\frac{1}{|G|}\sum_{n \in G}(deg(n))}{\frac{1}{|G|}\sum_{n \in G}(\frac{1}{|N(n)|}\sum_{m \in N(n)} deg(m))}$, with the average being taken across all nodes n in the network G, and where $N(n)$ is the neighborhood of node n. This value corresponds to the value of (average node degree/average average degree of node's neighbors) in the graph. If the networks are too large, consider a reasonable (what constitutes "reasonable" is up to your judgment) subsample of the nodes. Only consider outgoing edges when computing degree, and only consider neighbors to whom node n has an outgoing edge to that neighbor. If a node has out-degree of 0, then assume the average degree of its neighbors is 0.

Compute the value above for each of the following social networks:

- https://snap.stanford.edu/data/com-Orkut.html
- https://snap.stanford.edu/data/com-LiveJournal.html
- https://snap.stanford.edu/data/soc-Slashdot0811.html
- https://snap.stanford.edu/data/com-DBLP.html
- https://snap.stanford.edu/data/email-Enron.html
- https://snap.stanford.edu/data/com-Youtube.html
- https://snap.stanford.edu/data/soc-Epinions1.html
- https://snap.stanford.edu/data/wiki-Talk.html

You should use the first dataset in the 'Files' table on each page. Note that some of these graphs are directed, and some are undirected, as indicated by 'ungraph', which you should account for when building the graphs.

As a **bonus question (just for fun, no extra credit)**, for each network, plot the average of $\frac{deg(n)}{\frac{1}{|N(n)|}\sum_{m \in N(n)} deg(m)}$ against sample size (number of nodes n in the subsample). That is, for samples of varying sizes, compute this average for each sample on the y axis, and plot the sample size on the x axis.

You should also turn in to Canvas a script named '*Lastname_Firstname_ps4.py*', with your name substituted in place of '*Lastname_Firstname*', in addition to the rest of your problem set.

Solution by Khalid Tawil:

Network	Avg Node Degree	Avg Avg Neighbor Degree	Answer
email-Enron.txt	10.026	236.563	0.0424
soc-Epinions1.txt	7.907	49.289	0.1604
soc-Slashdot0811.txt	11.447	92.588	0.123
wiki-Talk.txt	34.02	1397.276	0.0243
com-dblp.ungraph.txt	6.618	18.484	0.358
com-youtube.ungraph.txt	5.264	652.094	0.00808
com-lj.ungraph.txt	17.346	74.562	0.232
com-orkut.ungraph.txt	76.281	350.534	0.218

Figure 1: Network A

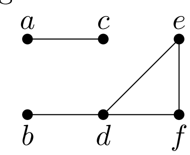

Figure 2: Network B